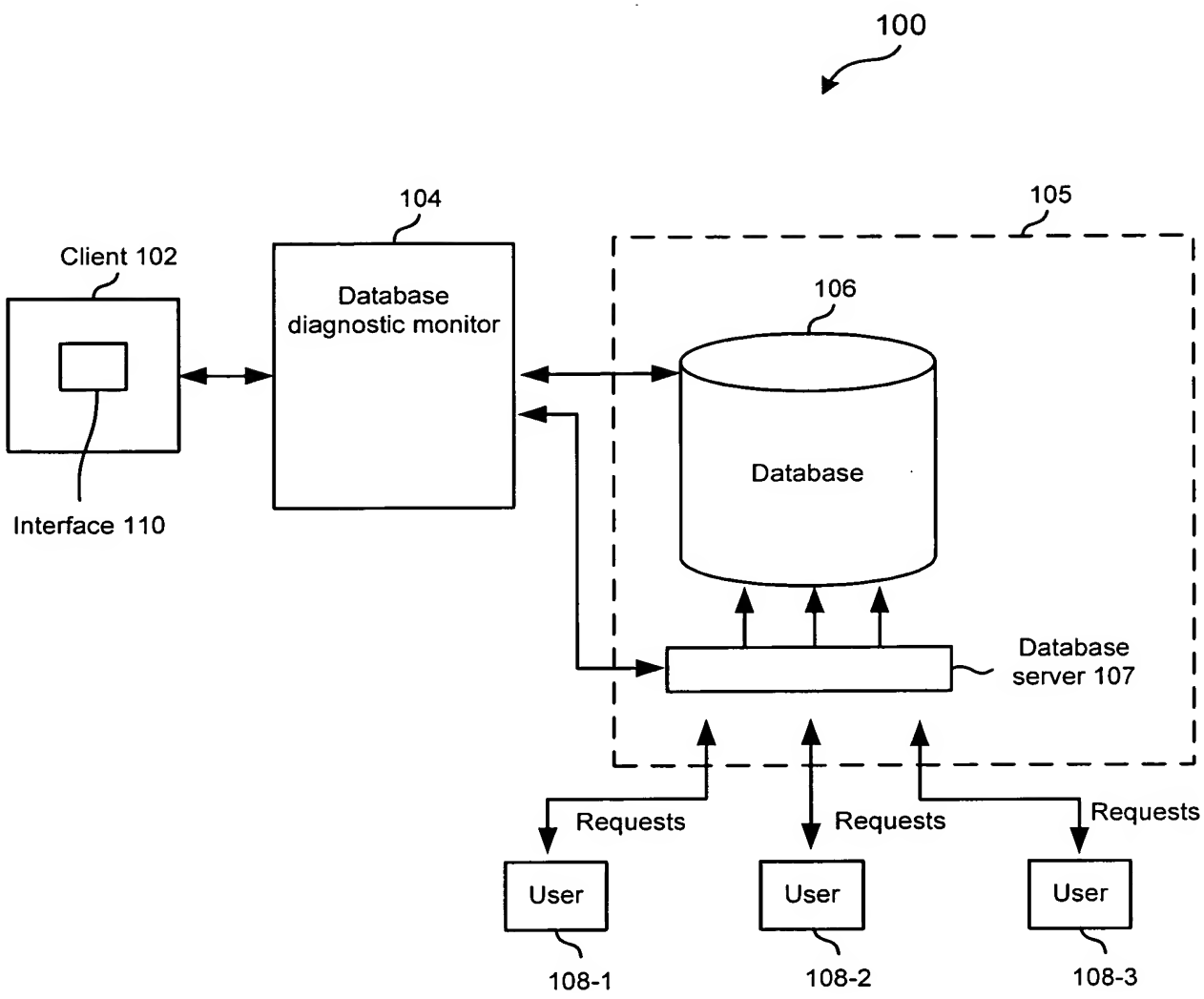

100
104
105
Client 102
Database diagnostic monitor
106
Database
Interface 110
Database server 107
Requests
Requests
Requests
User
User
User
108-1
108-2
108-3

*Fig. 1*

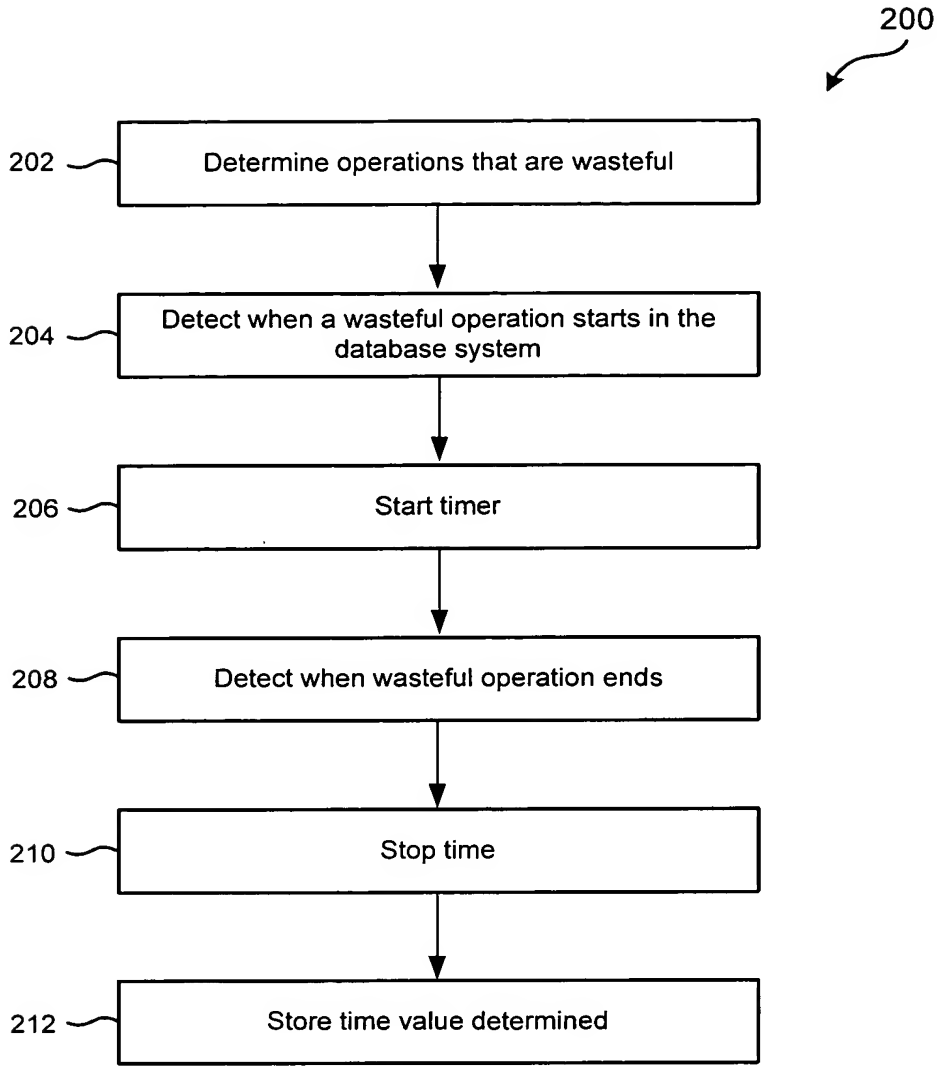

200
202
Determine operations that are wasteful
204
Detect when a wasteful operation starts in the database system
206
Start timer
208
Detect when wasteful operation ends
210
Stop time
212
Store time value determined

*Fig. 2*

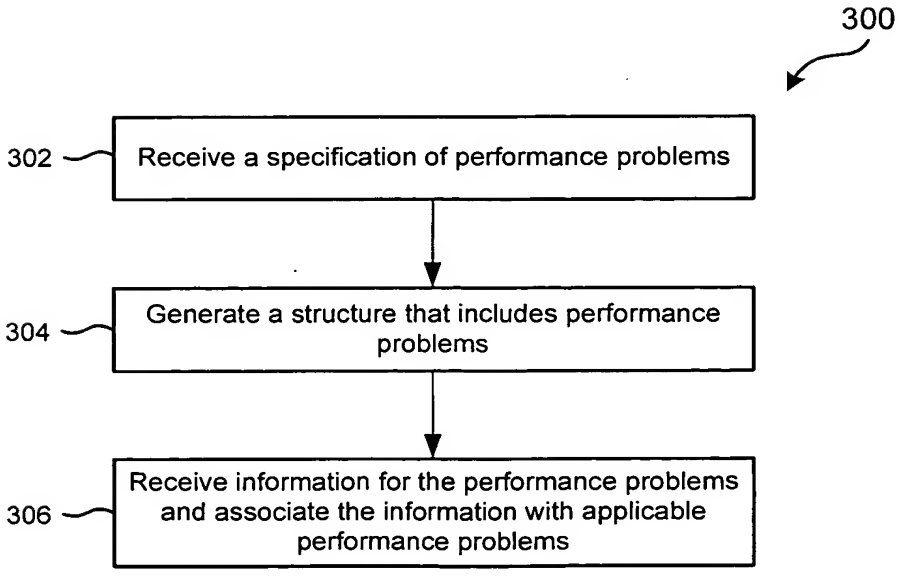

300
302
Receive a specification of performance problems
304
Generate a structure that includes performance problems
306
Receive information for the performance problems and associate the information with applicable performance problems

*Fig. 3*

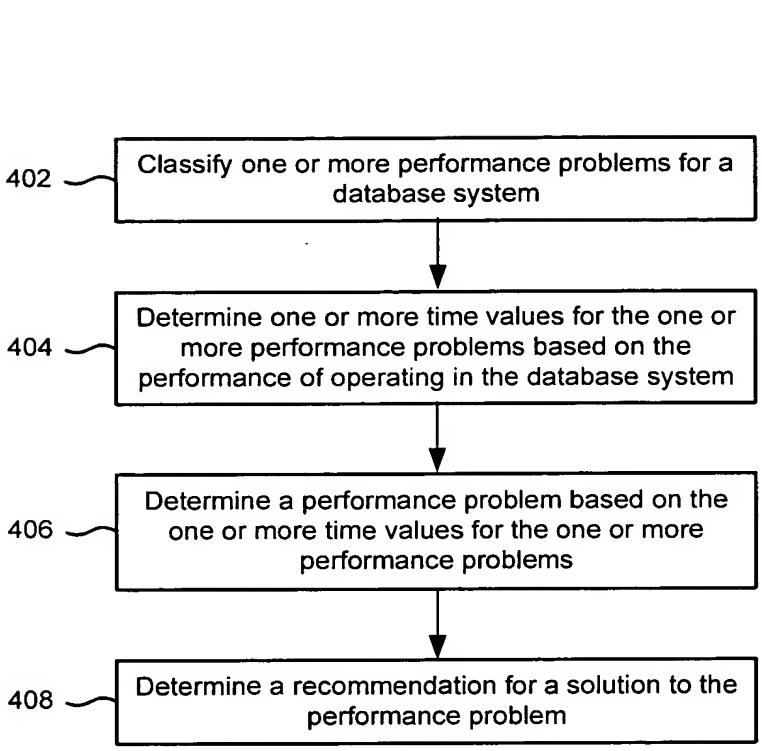

400
402
Classify one or more performance problems for a database system
404
Determine one or more time values for the one or more performance problems based on the performance of operating in the database system
406
Determine a performance problem based on the one or more time values for the one or more performance problems
408
Determine a recommendation for a solution to the performance problem

*Fig. 4*

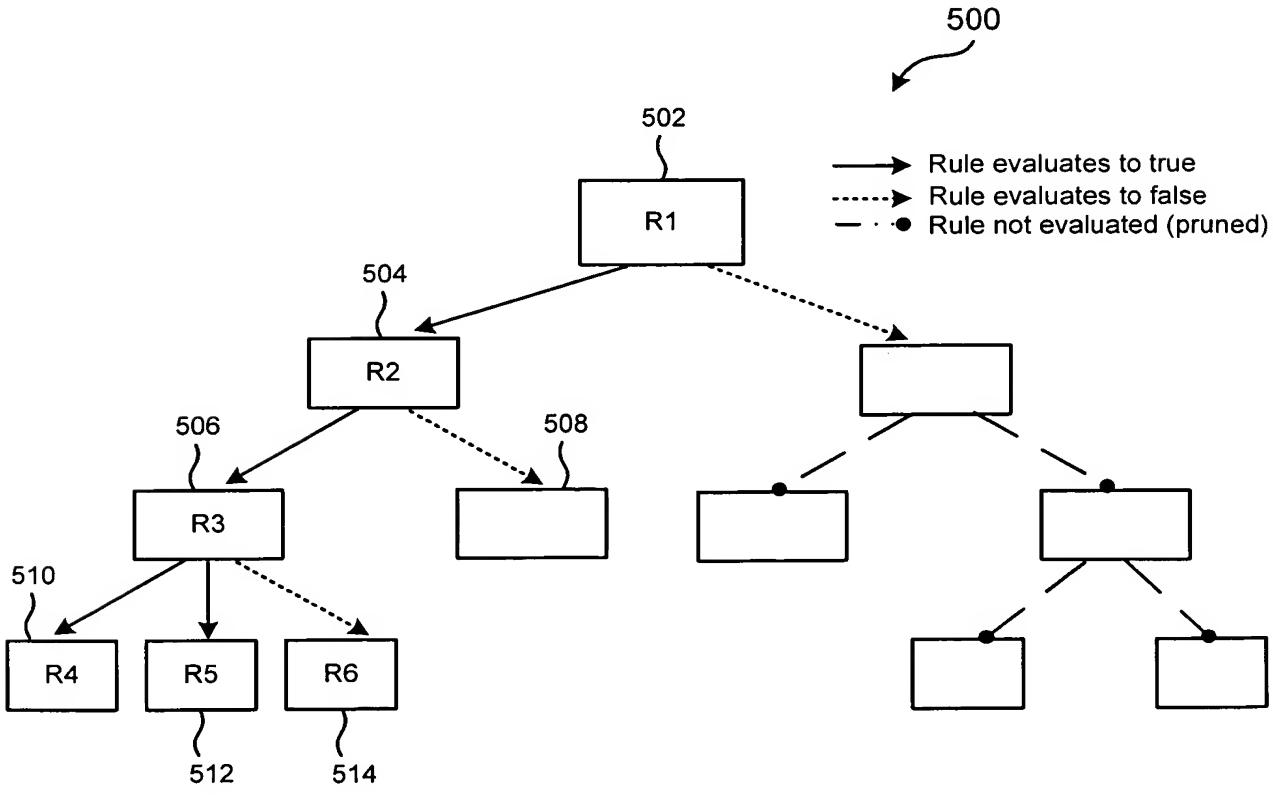

500
502
R1
Rule evaluates to true
Rule evaluates to false
Rule not evaluated (pruned)
504
R2
506
508
R3
510
R4
R5
R6
512
514

*Fig. 5*

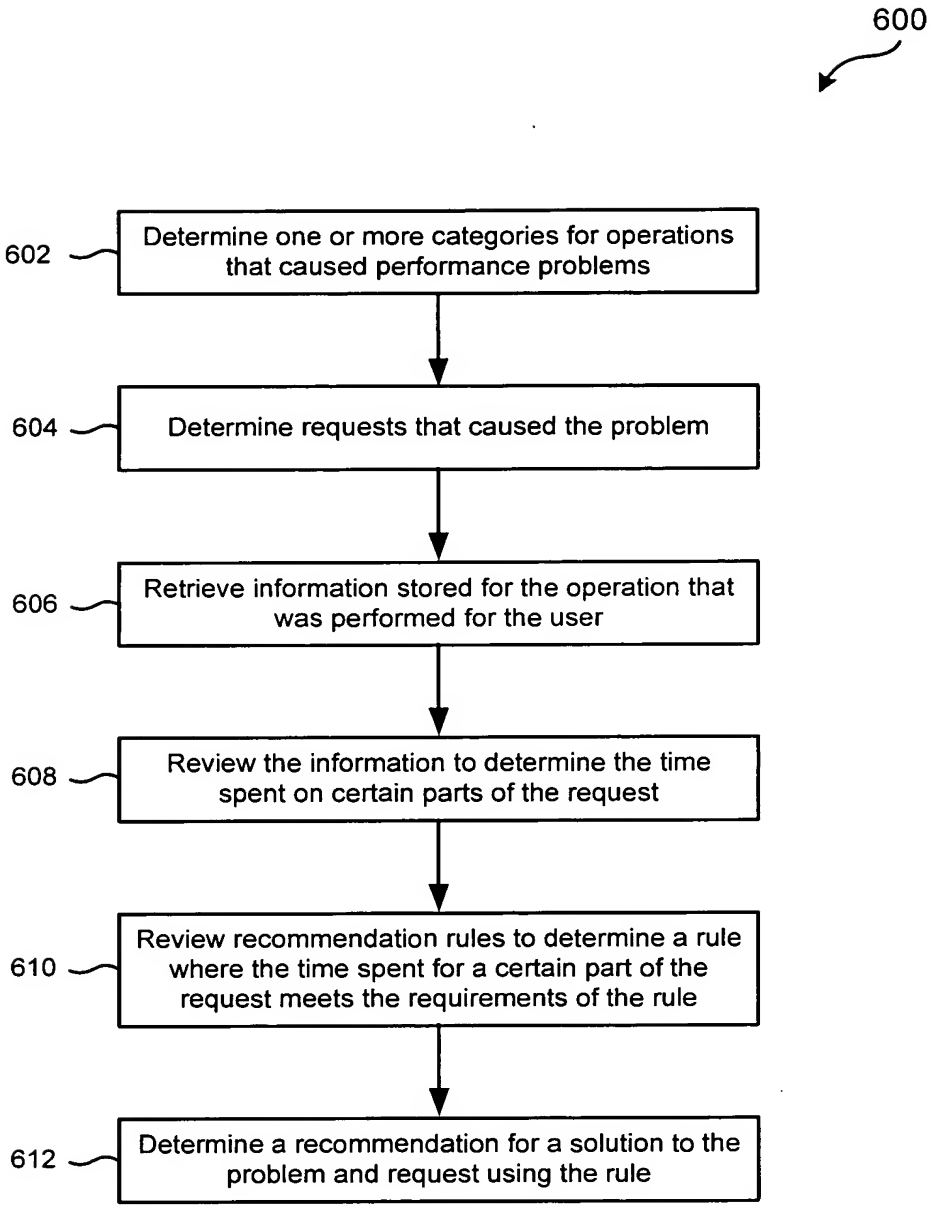

600
602
Determine one or more categories for operations that caused performance problems
604
Determine requests that caused the problem
606
Retrieve information stored for the operation that was performed for the user
608
Review the information to determine the time spent on certain parts of the request
610
Review recommendation rules to determine a rule where the time spent for a certain part of the request meets the requirements of the rule
612
Determine a recommendation for a solution to the problem and request using the rule

*Fig. 6*

Example 2: ADDM report

```
DBMS_ADVISOR.GET_TASK_REPORT('BB')
---------------------------------------------------------------------------
          DETAILED ADDM REPORT FOR TASK 'bb' WITH ID 16
          ---------------------------------------------

            Analysis Period: 30-MAY-2003 from 10:27:57 to 10:31:03
          Database ID/Instance: 1/1
               Snapshot Range: from 9 to 10
                Database Time: 1582 seconds
         Average Database Load: 8.5 active sessions

---------------------------------------------------------------------------
                                                          704-1
702-1  FINDING 1: 13% impact (201 seconds)
       ----------------------------------
       A hot data block with concurrent read and write activity was found. The block
       belongs to segment "RWBOLTON.TAB_BBW_DATABLOCK_I" and is block 70 in file 3.

706-1  RECOMMENDATION 1: Application Analysis, 13% benefit (201 seconds)
707-1  ACTION: Investigate application logic to find the cause of high
          concurrent read and write activity to the data present in this block.
          RELEVANT OBJECT: database block with object# 40984, file# 3 and
          block# 70
       RATIONALE: The SQL statement with SQL_ID "4vxy8fv4y3dhd" spent
          significant time on "buffer busy waits" for the hot block.
          RELEVANT OBJECT: SQL statement with SQL_ID 4vxy8fv4y3dhd
708-1     UPDATE TAB_BBW_DATABLOCK SET REC_ID = :B3+:B2+1 WHERE REC_ID = :B1
       RATIONALE: The SQL statement with SQL_ID "90n4zy8h6375p" spent
          significant time on "buffer busy waits" for the hot block.
          RELEVANT OBJECT: SQL statement with SQL_ID 90n4zy8h6375p
          UPDATE TAB_BBW_DATABLOCK SET REC_ID = :B3 WHERE REC_ID = :B2+:B1+1

710-1  SYMPTOMS THAT LED TO THE FINDING:
          Wait class "Concurrency" was consuming significant database time. (24%
          impact [375 seconds])
704-2
702-2  FINDING 2: 13% impact (201 seconds)
       ----------------------------------
       Read and write contention on database blocks was consuming significant
       database time.

706-2  RECOMMENDATION 1: Schema, 13% benefit (201 seconds)
707-2  ACTION: Consider hash partitioning the INDEX
          "RWBOLTON.TAB_BBW_DATABLOCK_I" with object id 40984 in a manner that
          will evenly distribute concurrent DML across multiple partitions.
          RELEVANT OBJECT: database object with id 40984
       RATIONALE: The UPDATE statement with SQL_ID "4vxy8fv4y3dhd" was
          significantly affected by "buffer busy waits".
          RELEVANT OBJECT: SQL statement with SQL_ID 4vxy8fv4y3dhd
708-2     UPDATE TAB_BBW_DATABLOCK SET REC_ID = :B3+:B2+1 WHERE REC_ID = :B1
       RATIONALE: The UPDATE statement with SQL_ID "90n4zy8h6375p" was
          significantly affected by "buffer busy waits".
          RELEVANT OBJECT: SQL statement with SQL_ID 90n4zy8h6375p
          UPDATE TAB_BBW_DATABLOCK SET REC_ID = :B3 WHERE REC_ID = :B2+:B1+1

710-2  SYMPTOMS THAT LED TO THE FINDING:
          Wait class "Concurrency" was consuming significant database time. (24%
          impact [375 seconds])
```

*Fig. 7A*

Example 2: ADDM report (continued)

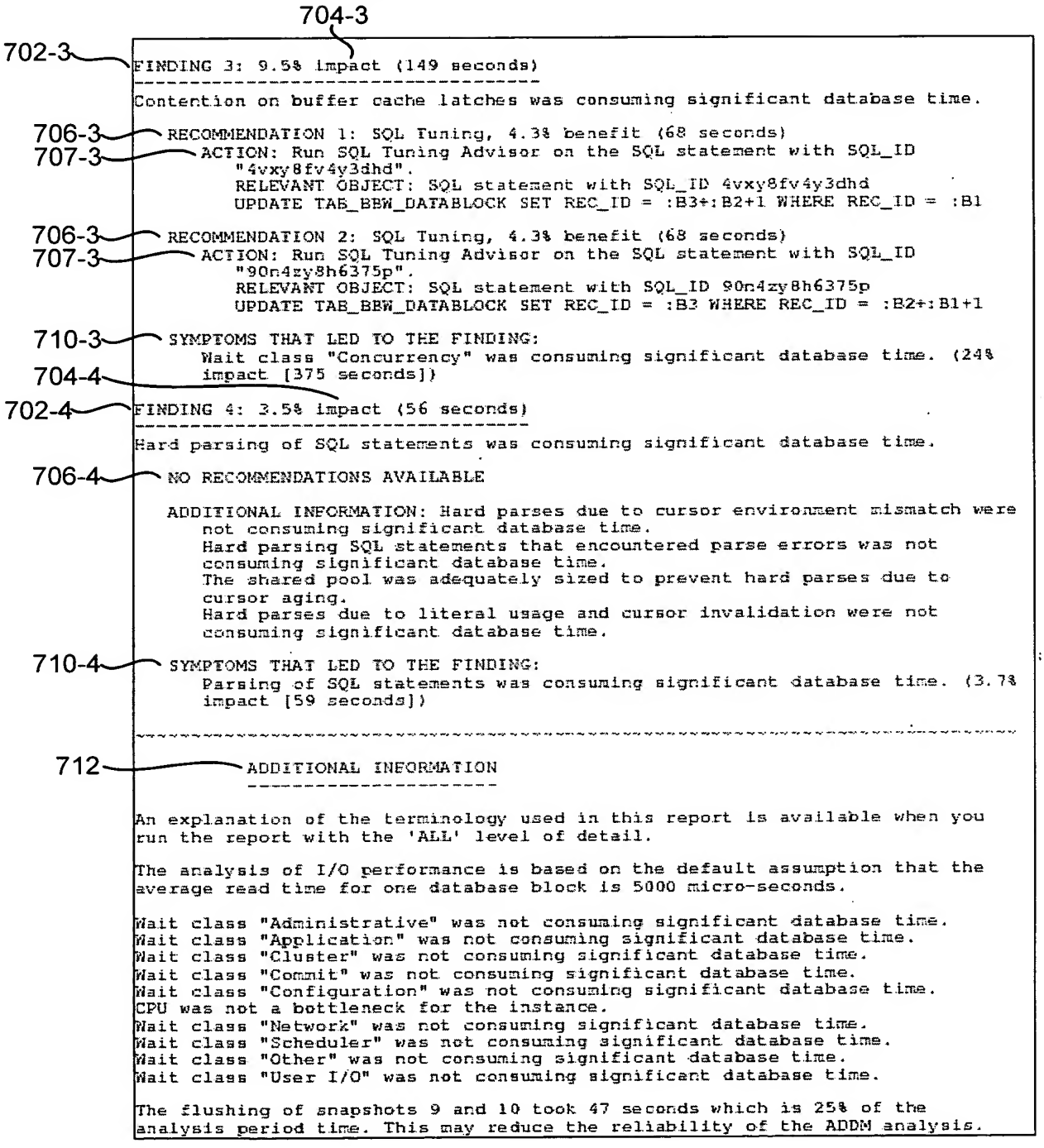



*Fig. 7B*

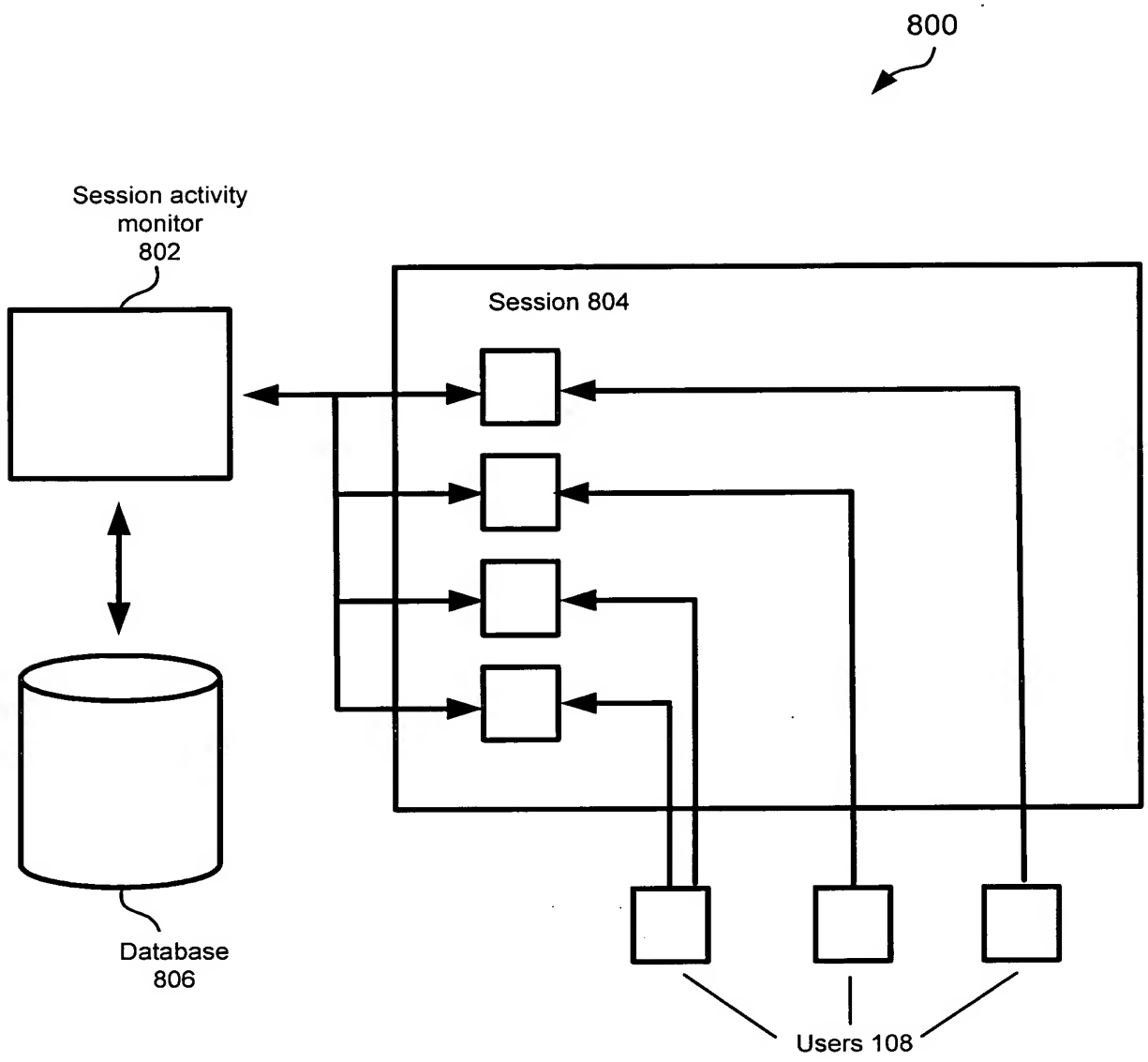

800
Session activity monitor 802
Session 804
Database 806
Users 108

*Fig. 8*

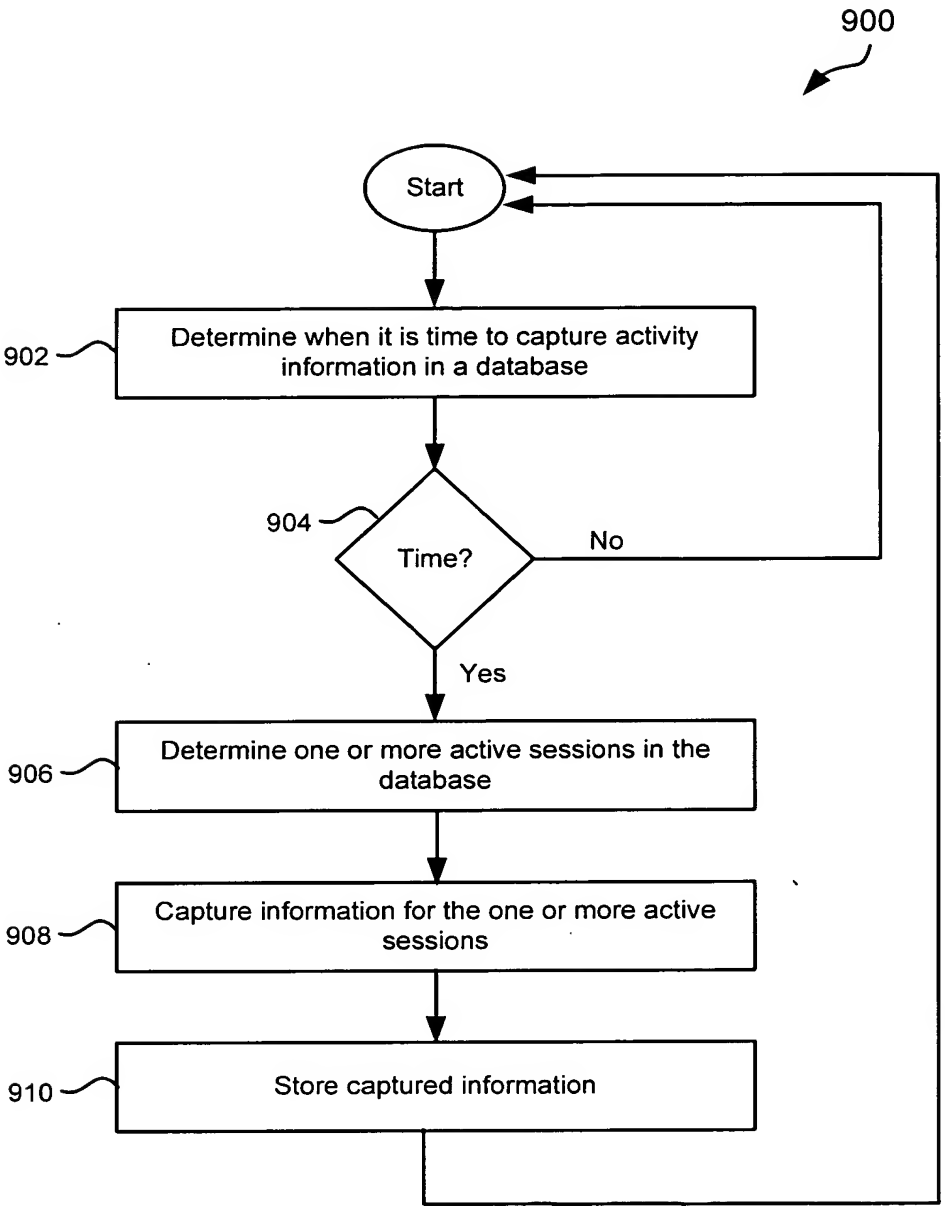
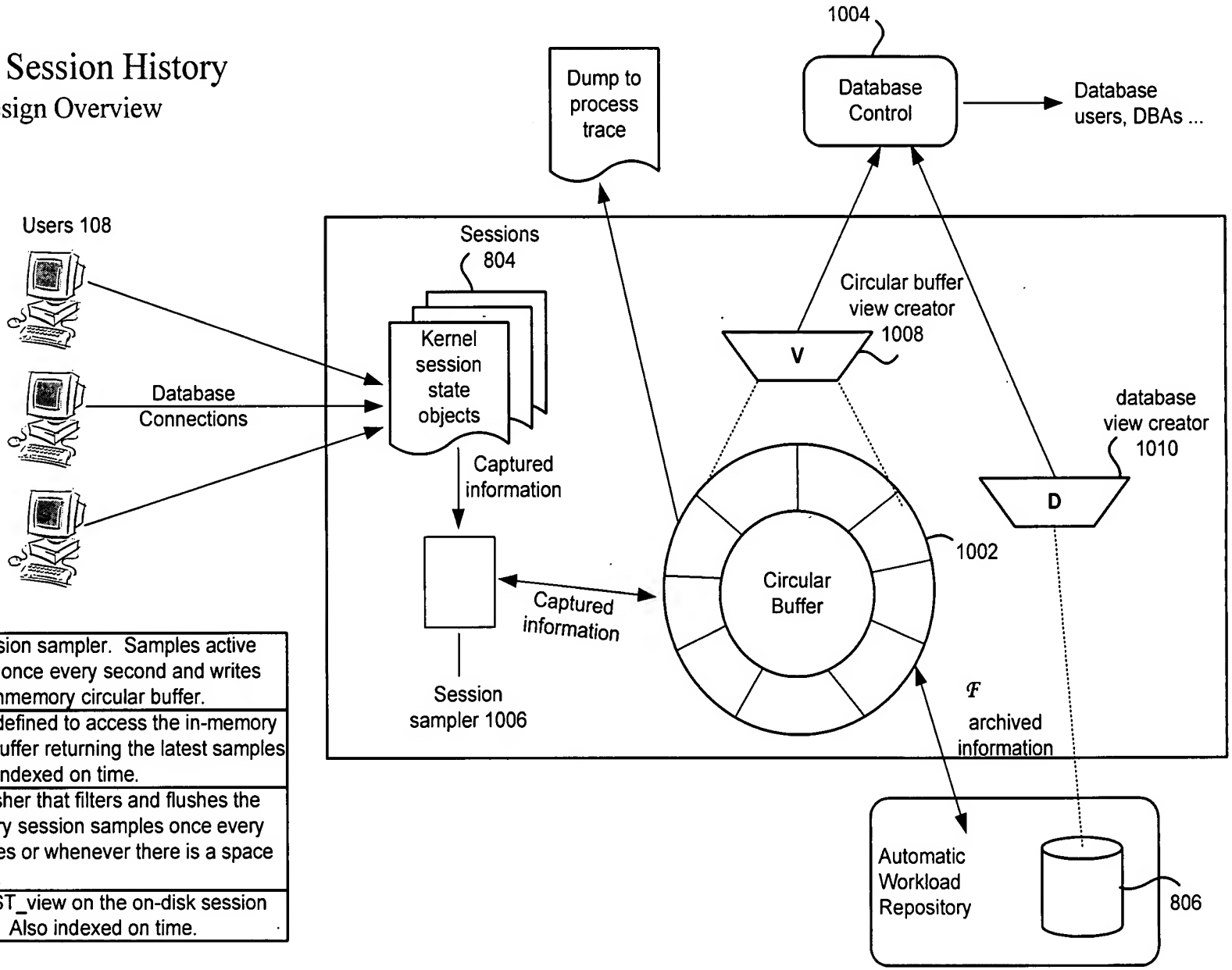

900
Start
902
Determine when it is time to capture activity information in a database
904
Time?
No
Yes
906
Determine one or more active sessions in the database
908
Capture information for the one or more active sessions
910
Store captured information

*Fig. 9*

# Active Session History
## Design Overview




| | |
|---|---|
| **S** | ASH session sampler. Samples active sessions once every second and writes into the inmemory circular buffer. |
| **V** | V$ view defined to access the in-memory circular buffer returning the latest samples first and indexed on time. |
| **F** | ASH Flusher that filters and flushes the in-memory session samples once every 30 minutes or whenever there is a space pressure. |
| **D** | DBA_HIST_view on the on-disk session samples. Also indexed on time. |

*Fig. 10*

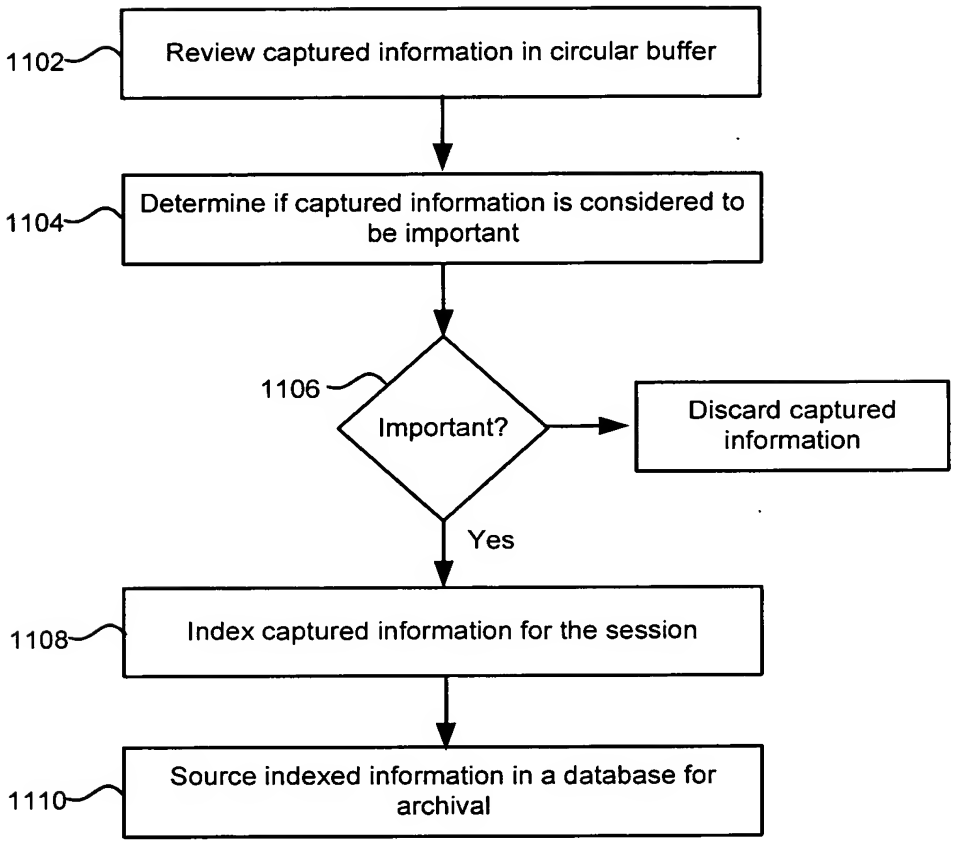

1102
Review captured information in circular buffer
1104
Determine if captured information is considered to be important
1106
Important?
Discard captured information
Yes
1108
Index captured information for the session
1110
Source indexed information in a database for archival

*Fig. 11*

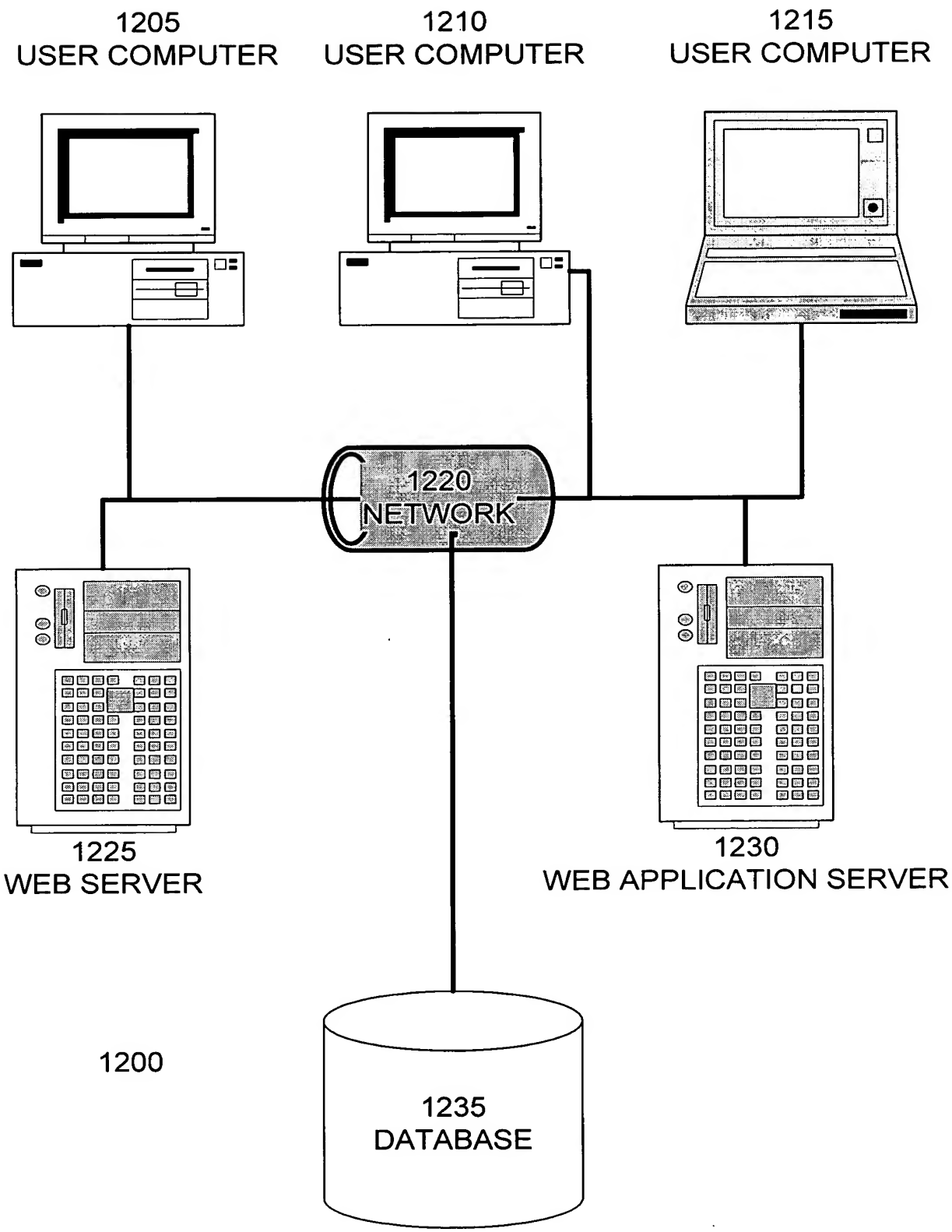

1205
USER COMPUTER
1210
USER COMPUTER
1215
USER COMPUTER
1220
NETWORK
1225
WEB SERVER
1230
WEB APPLICATION SERVER
1200
1235
DATABASE

FIG. 12